model. However, in such embodiments, the object detector **202** may still be configured to determine a central point or region for each object, and to detect changes in pose of the two objects relative to one another. This may include e.g. changes in relative separation and angle of the central points for each object. The changes in angle with respect to the two centre points of each object may be tracked by fitting a line between the centre points and tracking a rotation of the fitted line relative to some axis (e.g. the x-axis). It may be useful to use machine learning to perform the object identification, as such methods may be more robust and less reliant on e.g. a filtering operation that requires some prior knowledge of the object being held by the user. Provided that the machine learning model has been trained on a sufficient number of different objects, it should be possible to detect an object being held by the user as a video games controller.

[0070] In some embodiments, contour detection may be performed in a deterministic manner (i.e. without the use of machine learning), with the pose of the objects being detected via machine learning. For example, the object detector **202** may be configured to provide the object pose detector **203** with the contour(s) detected for a given object, and the machine learning model may be trained to determine a respective pose of the object based on the contour(s) input to the trained model.

[0071] In some embodiments, the object pose detector **203** is configured to detect whether the distance between points representative of the centre of each object is less than a threshold distance. The threshold distance may correspond to the length of one or each object. For example, if the user is holding two bananas, the object pose detector **203** may be configured to detect whether the distance between the centres is less than the length of one (or each) banana. In response to a positive determination, the object pose detector **203** may be configured to provide an input to the user input generator **204**, indicating that the distance is less than the threshold distance. In response to receive this input, the user input generator **204** is configured to generate a user input corresponding to a change in control mode. For example, a detection of two bananas being positioned end to end may result in a reverse command being generated.

[0072] It will be appreciated that, in some embodiments, a user may hold more than two objects in each hand, and therefore each of these objects and their respective poses will need to be detected. The detection of these objects and their poses may be performed as above with e.g. a greater number of contours and corresponding centre points being detected for each object. The motion of each object relative to the other objects may be used to generate user inputs, wherein each user input may be different depending on which of the objects has been moved, and the nature of the movement.

[0073] In some embodiments, the object pose detector **203** may comprise a machine learning model that has been trained to perform six-dimensional pose estimation for each non-luminous object detected by the object detector **202**. The machine learning model may comprise a convolutional neural network that has been trained for such purposes. A non-limiting example of such a machine learning model is the 'PoseCNN' model (see: 'PoseCNN: A convolutional neural network for 6D Pose Estimation in Cluttered Scenes', Y. Xiang et al, 26 May 2018, p. 1-10'). In such embodiments, the detected pose of at least one object may used as a higher dimensional controller, with motion in one or more dimensions being used to generate different respective user inputs. For example, rotation in the roll, pitch and yaw axes may each correspond to a different respective user input. Rotation about the roll axis may correspond to steering, while rotation about the yaw axis may correspond to acceleration/deceleration; rotation in about the yaw axis may correspond to e.g. a change in viewpoint. Similarly, translation in the x, y and z axes may each be assigned to different user inputs (the rotation and translation axes provided six dimensions in which the objects can move in).

[0074] In examples where a user is detected as holding two objects, the pose detector may be configured to detect the six-dimensional pose of each object in the obtained images. The user input generator **204** may be configured to generate different user inputs based on the changes in pose of one or both objects.

[0075] In embodiments where machine learning is used for detecting the pose(s) of the object(s) being held by the user, the object detector **202** may be configured to detect a plurality of keypoints for each non-luminous object being held by the user. The user input generator **204** may be configured to generate a user input based on a detection of the distance between at least some of the keypoints for each object being detected at greater or less than a threshold distance. For example, the object detector **202** may detect that a user is holding a cup, and that a saucer is also visible in the obtained images. In response to detecting the cup as coming into contact with the saucer (based on a detected proximity of corresponding keypoints), the user input generator **204** may generate a 'pause' command that is transmitted to the CPU of the video game playing device.

[0076] In some examples, it may be that two different users are each holding one or more non-luminous objects. For example, two users may each be holding a banana in a respective hand. To enable each of these users to participate in a multiplayer video game session, the system may further comprise a user identification unit operable to associate each non-luminous object with a different respective user. The user identification unit may be configured to associate a given object with a given user based on at least one of a relative (i) distance and (ii) depth associated with the object exceeding a threshold value. It may be for example, that two objects corresponding to bananas are identified in a left and right region of the obtained images, and that the left-most banana is assigned to the first player and the right-most to the second player. The threshold distance and/or depth may be used to distinguish between the case where a single user is holding two objects.

[0077] In more complex examples, it may be that the user identification unit is configured to detect the users in the obtained images. In such examples, a given object may be associated to a given player based on the position and/or depth of that object relative to a detected position and/or depth of the corresponding player. For example, a given object may be associated to the player that is detected as being closest to it. As will be appreciated, in order to estimate a depth of the object and player, the obtained images may need to include a depth image, or a stereoscopic image from which a depth image can be estimated. In some cases, it may be known in advance that a user intends to use e.g. a fruit as a controller, and the relative to depth of the player and fruit may be estimated based on the size of the fruit in the obtained images.